Wiley Encyclopedia of Electrical and Electronics Engineering
DOI:10.1002/047134608X; ISBN:9780471346081

Range Images

Yonghuai Liu*, Department of Computer Science, Aberystwyth University, UK
Peter Yuen, Centre for Electronics Warfare, Cranfield University, UK
Yanwei Pang, School of Electrical and Information Engineering, Tianjin University, China
Yitian Zhao, Cixi Institute of Biomedical Engineering, Chinese Academy of Sciences, Ningbo, China
Paul L. Rosin, School of Computer Science & Informatics, Cardiff University, UK

Abstract

While range images can easily be captured through readily available commercial off-the-shelf range cameras/sensors/finders nowadays, it is not immediate for the user to appreciate and understand the technological challenges involved in range imaging. This article gives an overview of range imaging techniques with an aim to let the reader better understand how the difficult issue, such as the registration of overlapping range images, can be approached and solved. This article firstly introduces the characteristics of range images and highlights examples of 3D image visualizations, associated technical issues, applications and the differences of range imaging with respect to the traditional digital broadband imaging. Subsequently, one of the most popular feature extraction and matching methods, the signature of histograms of orientations (SHOT) method, is then outlined. However, the "matched" points generated by SHOT usually generates high proportion of false positives due to various factors such as imaging noise, lack of features and cluttered backgrounds. Thus the article discusses more about image matching issues particularly to emphasize how the widely employed range image alignment technique, the random sample consensus (RANSAC) method, is compared with a simple yet effective technique based on normalized error penalization (NEP). This simple NEP method utilizes a strategy to penalise point matches whose errors are far away from the majority. The capability of the method for the evaluation of point matches between overlapping range images is illustrated by experiments using real range image data sets. Interestingly enough, these range images appear to be easier to register than expected. Finally, some conclusions have been drawn and further readings for other fundamental techniques and concepts have been suggested.

Keywords : Range images; applications; imaging noise; visualization, feature extraction and matching; point match evaluation; underlying transformation.

*Corresponding author. Email: yyl@aber.ac.uk; Tel: +44 1970 621688; Fax: +44 1970 628536

1 Introduction

The advancements in electro-optics, electronics, mechanics and control technologies enable the advent of modern laser scanning systems to facilitate high precision 3D model imaging applications. There are generally three main steps involved in modern 3D range imaging: (i) surface parts reconstruction, (ii) image matching (registration) and (iii) image stitching. The first step can be achieved by techniques such as laser scanners, structured light projection or multi-view reconstruction. Sophisticated optical design allows the generation and control of active light sources such as laser beams, custom light patterns etc., for the illumination of objects of interest without interference by natural light such as the solar irradiance. The advances in electronics enables the effective and efficient data acquisitions, data transmission, data storage and signal processing of reflections of light from the object of interest. The advancement in mechanical control enables accurate controls of the scanner components to allow simultaneous acquisition of wide field-of-view of the scene in both depth (range) and intensity (Figure 1).

The second step of image matching/registration is to position all surface parts in a common coordinate system. This is necessary because of the limited field-of-view in most laser scanning systems (range cameras), thereby several (or even hundreds of) images taken at different viewing points are needed to cover the whole scene. If two range images scan over a common part of the scene, then the images of these two scans are termed as overlapping. All range images are normally recorded in the local coordinate of the laser scanning system. To construct a full model of the object and/or fuse the geometrical and optical information, all these overlapping range images are needed to be aligned in a single global coordinate system. This process is termed as image registration. Range image registration has two goals: one is to establish correspondences

between overlapping range images, the other is to estimate the underlying transformation that brings one range image into the best possible alignment with the other. Fixing either of these two goals renders the other trivial. However, in practice they are interwoven, thus making range image registration to be one of the most challenging tasks in range imaging. The 3D model can then be formed by stitching the registered surface parts in voxel space [1], or by direct mesh fusing methods [2]. Recent advancement in this area has been the employment of KinectFusion technique [3] together with the Microsoft Kinect sensor for tracking and mapping of indoor scenes, which involves processes such as data capture, image registration, image fusion and reconstruction of dense surfaces in the volumetric space and all these processes can be performed in real time.

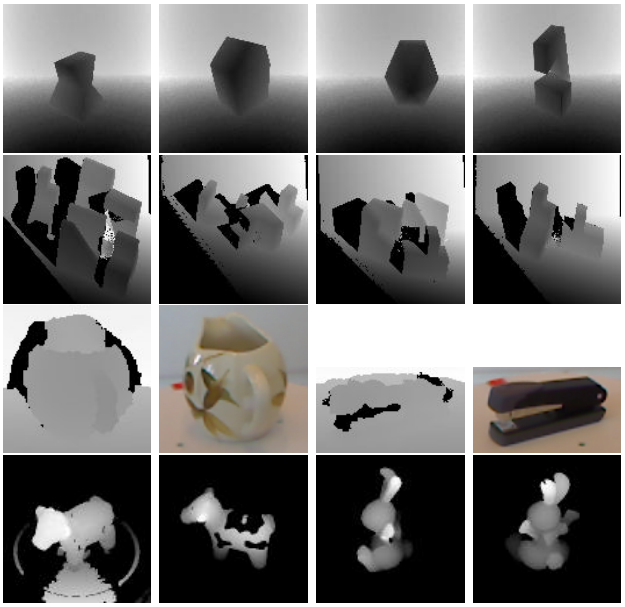

Figure 1: Examples of real range images and their corresponding intensity images, when available and explicitly stated. Top row from left to right: ptrain0, ptrain1, ptrain2, and ptrain3 captured by a Perceptron scanner [4]. Second row from left to right: atrain0, atrain1, atrain2, and atrain3 captured by an ABW scanner [4]. Third row from left to right: pitcher1, pitcher1 intensity image, stapler1, and stapler1 intensity image captured by a Kinect sensor [8]. Bottom from left to right: cow42, cow37, bunny120 and bunny60 captured using a Minolta Vivid 700 range camera [12].

1.1 Range images

A range image is a two dimensional rectangular array with each pixel representing the distance of a point on the surface of an object of interest with respected to either a reference point, or a reference plane, as defined by the range imaging device (range camera) or system. Each pixel is parameterized using three variables $(i, j, r(i, j))$ where i is the column index, j is the row index, and $r(i, j)$ is the distance of a point from the reference point/plane. Depending on the type of the imaging device the distance of a point can be measured with respect to either a reference point as in the perspective projection, or with respect to a reference plane in the case of orthographical projection system. For example, in the case of the Perceptron laser range finder [4, 5, 6, 7] the distance is measured with respected to a reference point, while the reference plane is used in the ABW range camera [4, 7] and the Kinect sensor [8]. These two categories of range imaging systems may be termed as perspective and orthographical range camera respectively.

1.1.1 Range image formation

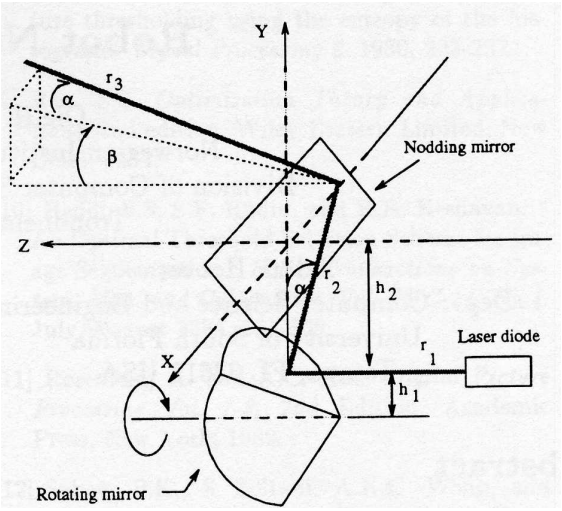

Figure 2: The operation principle of the Perception laser range finder [6].

Figure 2 illustrates the operation principle of the Perceptron laser range finder [6, 9] which employs a near-infrared laser of $810nm$ wavelength with a rotating mirror at r_1 distance away along the Z axis from the laser source and at zenith of α as illustrated in the figure. The rotating mirror is acted as the beam deflector in the X-Y plane. The laser beam is then directed over a distance of r_2 towards the nodding mirror which deflects the beam towards the scene of interest over a distance of r_3. When $\alpha = 0°$, the laser beam is defected from the rotating mirror along the Y axis. When $\alpha = 0°$ and $\beta = 0°$, the laser beam is deflected from the nodding mirror along the Z axis. The reflected laser beam from the scene is filtered,

and subsequently detected, by an avalanche photodiode light sensor. The distance of the laser beam travels, and thus the range of the object in the scene, is proportional to the phase difference between the transmitted and received laser signals. This phase difference is measured electronically. The pixel locations at (i, j) is related to the mirror angles (α, β), where β is the subtended angle between the X-Z plane and the plane passing through the Z axis and the laser beam $\mathbf{r}_3$ (please refer to Equation 1 in the next subsection). The phase difference is digitized into 12 bits yielding a measurement precision in the order of 2.0mm. A final circuit, which normally implemented as a look-up table, is designed to improve the accuracy of the computed range due to the variable laser energy reflected from different materials of the objects in the scene. This technique has an ambiguity of 2π for the determination of the phase difference. Thus, a good range imaging normally requires either to limit the variety/types and distances of objects in the scene or to employ external constraints to resolve this ambiguity.

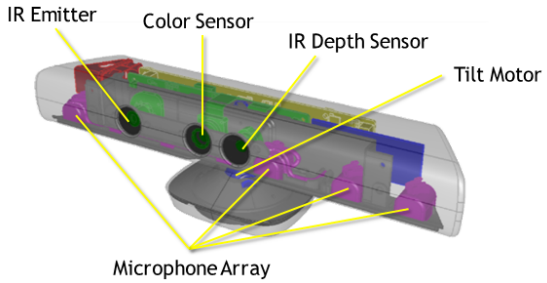

Figure 3: The main components of a Kinect sensor [10].

The Kinect sensor (Figure 3) has been one of the most popular 3D sensors in the market since it was released in 2010 [8, 11]. The Kinect measures the depth of scenes based on active triangulation [11] of the reflected light. It emits a pattern of infra-red (IR) light-words towards the scene of interest and a CMOS IR depth sensor is deployed for the detection of the reflected pattern. The IR Depth Sensor and IR Emitter form a binocular vision system, in which the received and emitted light-words that form pixels p and p' at positions of (i, j) and $(i + d, j)$ in their image planes can be matched along the same row j, thus to allow the identification of disparity $d = x'_1 - x_1$ of the pixels. The depth of the point P in the scene that reflects the light-word can be estimated through trigonometry as illustrated in Figure 4 : $r(i, j) = Bf/d$, where B is the distance between the light emitter and the depth sensor, and f is the focal length of the depth sensor. However, the light pattern in the same row should be distinctive enough to facilitate the matching of the

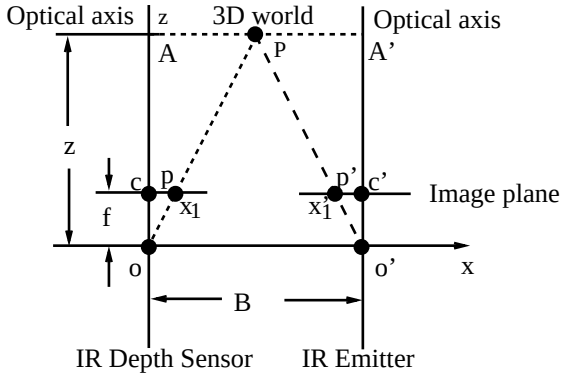

Figure 4: Illustrates the principle of the binocular vision system for the estimation of the depth z of the real world scene. The system consists of parallel optical axes, co-planar image planes and the same focal length of f for both the IR Depth Sensor (left) and the IR Emitter (right).

light words along the row. Such a constraint limits the measurement precision of the depth of the scene. The Kinect sensor operates at a resolution of 640×480 in the 32bit color stream, and 320×240 in the 16bit depth stream at 30fps respectively. It has a horizontal field of view of 57°, a vertical field of view of 43° and a depth range of 1.2m-3.5m with a resolution of 1cm at a distance of 2m.

1.1.2 3D Cartesian representation of range images

To visualize the range image it is necessary to port the detected depth in $(i, j, r(i, j))$ coordinates into 3D Cartesian representation (x, y, z) of the camera coordinate. This can be achieved through a function g which can be in any arbitrary units such as in millimeters. The function g can be estimated through a system calibration: $(x, y, z) = g(i, j, r(i, j))$ where g is a property of the imaging system. For example, the pixels $(i, j, r(i, j))$ $(0 \leq i, j \leq 511)$ of range images captured by perspective Perceptron laser range finder can be converted into Cartesian coordinates $(x(i, j), y(i, j), z(i, j))$ through the following relationships [4, 6, 7, 9]:

$$
\begin{aligned}
x(i,j) &= dx + r_3 \sin(\alpha) \\
y(i,j) &= dy + r_3 \cos(\alpha) \sin(\beta) \\
z(i,j) &= dz + r_3 \cos(\alpha) \cos(\beta) \\
r_1 &= (dz - h_2)/\delta \\
r_2 &= \sqrt{(dx)^2 + (h_2 + dy)^2}/\delta \\
r_3 &= (r(i,j) + r_0 - (r_1 + r_2))\delta \\
dx &= (h_2 + dy) \tan(\alpha)
\end{aligned}
$$

$$
\begin{aligned}
dy &= dz\tan(\theta + 0.5 * \beta) \\
dz &= -h_1(1.0 - \cos(\alpha))/\tan(\gamma) \\
\alpha &= \alpha_0 + H(255.5 - j)/512 \\
\beta &= \beta_0 + V(255.5 - i)/512
\end{aligned} \tag{1}
$$

where the specific values of h_1, h_2, γ, θ, α_0, β_0, H, V, r_0 and δ can be obtained through calibration [6] before the imaging. For images as shown in the top row of Figure 1, the values of these parameters are: $h_1 = 3.0$, $h_2 = 5.5$, $\gamma = \theta = 45°$, $\alpha_0 = \beta_0 = 0.0$, $H = 51.65$, $V = 36.73$, $r_0 = 830.3$, and $\delta = 0.20236$ respectively.

On the other hand, the pixels $(i, j, r(i, j))$ $(0 \leq i, j \leq 511)$ of range images captured by the orthographical ABW range camera can be converted into Cartesian coordinates $(x(i, j), y(i, j), z(i, j))$ as follows [7]:

$$
\begin{aligned}
x(i,j) &= (j - 255)(r(i,j)/\mathit{scal} + \mathit{offset})/|f_k| \\
y(i,j) &= (255 - i)/c(r(i,j)/\mathit{scal} + \mathit{offset})/|f_k| \\
z(i,j) &= (255 - r(i,j))/\mathit{scal}
\end{aligned}
$$

where the specific values of *offset*, *scal*, f_k and c can be obtained through calibrations before image acquisition. Similarly, the values of the parameters for the images shown in the second row of Figure 1 are: *offset* $= 785.410786$, *scal* $= 0.773545$, $f_k = -1610.981722$, $c = 1.4508$ for the first image while the second and third images have values of *offset* $= 771.016866$, *scal* $= 0.791686$, $f_k = -1586.072821$, $c = 1.4508$.

Note that in the case of the Kinect sensors, the orthographical-camera Cartesian relationship $((i, j, r(i, j))$-$(x(i, j), y(i, j), z(i, j)))$ in this system is different from that of the orthographical ABW range camera due to their different sensor sizes and scale factors [8]:

$$
\begin{aligned}
\mathit{xgrid} &= i + (\mathit{topleft}_x - 1) - \mathit{centre}_x \\
\mathit{ygrid} &= j + (\mathit{topleft}_y - 1) - \mathit{centre}_y \\
x(i,j) &= \mathit{xgrid} * r(i,j)/\mathit{constant}/MM\_Per\_M \\
y(i,j) &= \mathit{ygrid} * r(i,j)/\mathit{constant}/MM\_Per\_M \\
z(i,j) &= r(i,j)/MM\_Per\_M
\end{aligned}
$$

where $\mathit{centre}_x = 320$, $\mathit{centre}_y = 240$, *constant* $= 570.3$, and $MM\_PER\_M = 1000$. In the case of full frame images the i and j ranges from 1 to 640 and from 1 to 480, and $\mathit{topleft}_x = 1$ and $\mathit{topleft}_y = 1$ respectively. While the subset images as that shown in the third row of Figure 1, e.g. in the pitcher1 image, the i and j ranges from 1 to 110 and from 1 to 109, the $\mathit{topleft}_x = 262$ and $\mathit{topleft}_y = 191$ respectively. For the stapler1 image the i and j ranges from 1 to 118 and from 1 to 68, the $\mathit{topleft}_x = 250$ and $\mathit{topleft}_y = 240$ respectively. It is noted that the Kinect and all other range camera systems are capable of capturing both range and intensity images and thus they are intended for applications such as object modeling and recognition.

Figure 5: The figure shows the 3D models of soyabean and wheat plant reconstructed by using respectively 108 and 90 RGB images [14] with point clouds represented in *PLY* format [15]. The 3D models exhibit missing points (holes) in the plant structure and the clutter backgrounds such as the pot, soil and pot label etc. that cause ambiguity in the determination of neigbouring points.

One side effect for the transformation of the orthographical indexes (i, j) of the range images into the camera x and y Cartesian coordinates is the reduced accuracy in the distance representation due to the unity increments in the Cartesian coordinate system. The spatial and depth resolutions are both affected especially when the ranges between points are less than 1 unit. In contrast, the range images of the perspective range cameras transform the coordinates directly based on the imaging geometry, and thus the perspective system provides a more accurate 3D model than that of the orthographical range images.

Due to the limited field of view of the scanner and also because of the objects occlusion, multiple range images taken from different viewpoints are needed to fulfill the 3D modeling for the complete scene. It is noted from the above that all range images are captured with reference to the local scanner coordinate systems. Thus the images that are taken at different viewpoints are needed to be fused, and registered, into a single common global coordinate system.

While the range cameras such as the ABW [4, 7] and Kinect [8] which are capable to capture the depth information of a scene with respected to a reference plane, depth maps can also be generated from the

stereo system based on image matching [13]. This stems from the fact that the Cartesian coordinates of 3D models can be estimated from the motion of a feature within the multi-view images and that can be realised in two different steps of feature extraction and matching method: sparse and dense reconstruction [14]. The range imaging approach employs active light source for scene illumination while the stereo system utilizes ambient light. Consequently, the former achieves better result independent of environment factors while the latter suffers from illumination artifact or lack of texture with higher degree of missing points (holes) in the 3D reconstruction as shown in Figure 5.

In summary, three dimensional model of the scene can be reconstructed by using various sophisticated range camera systems as well as advanced image processing methods. In the latter case the point clouds can even be generated through the perspective view of the scene collected at different view angles. The range images, and thus their respective point clouds, are characterized by the noise, spatial and depth resolution that determine their fitness for specific applications. To echo this point here are several quotes about the nature of range images [16]: "Range images encode the position of surface directly. Therefore, the shape can be computed reasonably easy. Range images are a special class of digital images. Each pixel of a range image expresses the distance between a known reference frame and a visible point in the scene. Therefore, a range image reproduces the 3D structure of a scene. Range images are also referred to as depth images, depth maps, xyz maps, surface profiles and 2.5D images. Range images can be represented in two basic forms. One is a list of 3D coordinates in a given reference frame (cloud of points), for which no specific order is required. The other is a matrix of depth values of points along the directions of the x,y image axes, which makes spatial organisation explicit."

In [17], "Range imaging is the name for a collection of techniques that are used to produce a 2D image showing the distance to points in a scene from a specific point, normally associated with some type of sensor device. The resulting image, the range image, has pixel values that correspond to the distance. If the sensor that is used to produce the range image is properly calibrated, the pixel values can be given directly in physical units, such as meters."

In [18], "A range image is, in principle, a regular (i, j) grid of points with a depth at every pixel and a projection procedure for taking a point in 3D and mapping (projecting) it into to grid coordinates. In practice, the projection may only be approximately known, and there may be some irregularity to the arrangement of the points (e.g., every other line may be shifted due to interlacing). For this reason, we represent a range image as a grid of points in 3D (i.e., an (i, j) grid with an x, y, and z coordinate at every pixel) that is roughly observed with a view direction looking down the z-axis."

1.2 Applications

Since the depth information in range images contains geometrical information with physical units, they are more attractive for applications such as measurement, visualization and decision making over the broad band digital images. This section outlines some examples of range images for practical applications.

1.2.1 Object modeling

One of the most mature areas for the application of range imaging technologies has been the digitization and archiving of historical relics for conservation and maintenance purposes. These historical relics are normally fragile and degrading in color and structure over time. Thus, there is an urgent need to archive a digital representation of such relics for storage, repair and also for remote tourism. Several range finding techniques, including photogrammetry, structured light triangulation, time of flight and interferometry had been used in [19] for scanning Michelangelo's statues. However, it took one year of thirty faculty members of staff and students to scan ten of these statues in Italy! A single range image data set consists of as many as 2 billion polygons and 7000 color images of David!

Another sensing system known as the flying laser range sensor (FLRS) which incorporated a movable platform, had been designed [20] for digitizing large structure, such as the Bayon temple located at the centre of the Angkor-Thom in the kingdom of Cambodia. The temple measures 150 meters long on all sides and up to 45 meters tall. During the operation the FLRS was suspended by a balloon remotely controlled from the ground to reach the height and the occluded parts of the temple. Any distortions of the imaging due to the movement of the balloon had been removed using either the ground based imaging data or through a structure from motion based method. Various sensors had been deployed for capturing different aspects of data such as the range and their appearances. A single constructed model of the temple consists of 20000 range images with a total file size of about 200GB.

1.2.2 Simultaneous localization and mapping for robot autonomous navigation

Range imaging has been one of the key techniques to facilitate navigation of robots. Feedback information, such as the environment (scene), the movement

and posture of the robot are essentially needed for the navigation of the robot autonomously. This kind of information is continuously needed during maneuver of the robot to allow successful robotic operations such as grasping, moving and path planning. Laser range finder had been used in [6] for two types of motion planning: (1) physical safety: how to maneuver the robot from one point to another without collision to obstacles and (2) visual safety: which involves viewing plan to explore unknown/unseen areas given the constraints of the range finder and the obstacles in the environment. When multiple candidates present in the scene the one with maximum unknown territory will be selected. Based on such principles, a robot can navigate from one location to another and fulfill its goals successfully.

Experimentally a laser sensor had been mounted [21] at the end effector of a robot for navigation around a household environment: a kitchen in this case. The scanned images were pre-processed, resampled, segmented and various features were extracted for classifications and recognitions. Objects in the scene such as cupboards, tables, drawers and shelves had been identified and localized. The state of these objects, whether they were closed or opened, had also been recognized. A semantic map of the environment had been built based on all this information to facilitate the development of household assisted robots with autonomous navigation and tasking capabilities.

1.2.3 Quality assurance

The 3D models reconstructed from range images have been utilised for direct comparison between the CAD models of the components/products of an industrial system for quality assurance. Any discrepancy found may imply the presence of defective components/products which are needed to be replaced or singled out. Two different models of range cameras had been developed in [22] to capture the 3D data of underwater oil and gas pipes. The type 1 range camera was implemented by placing the laser source at the top of the linear stage at a fixed **elevation** angle. The laser beam was directed from the air into the water. The type 2 range camera placed the light generator in front of the laser behind the front glass. The scanned range data were used to visualize and to analyse whether the geometry of the pipes was according to the design and with a scope of defects localisation such as the identification of dents and damage/corrosion in the pipe (Figure 6).

Another example was the implementation of the Technical Arts 100X white scanner [23] for the inspection of a loaded printed circuit board. The scanner is capable for scanning dense and high resolution depth data at a resolution of $50\mu m$, with a maximum view window of 12.5mm in the horizontal direction and 10mm in the vertical direction. In the experiment the 3D data was segmented using a region growing method for the extraction and comparison of various features against those from the CAD models. The task was to verify parameters such as dimensions, co-planarity and location of features etc. In the solder joint inspection task the scanned data is registered with the designated CAD board with errors of less than $\pm 0.1mm$ in the case of good solder joints. The experiment had shown effective differentiation between good and bad joints by exploiting the volume and texture features for classification.

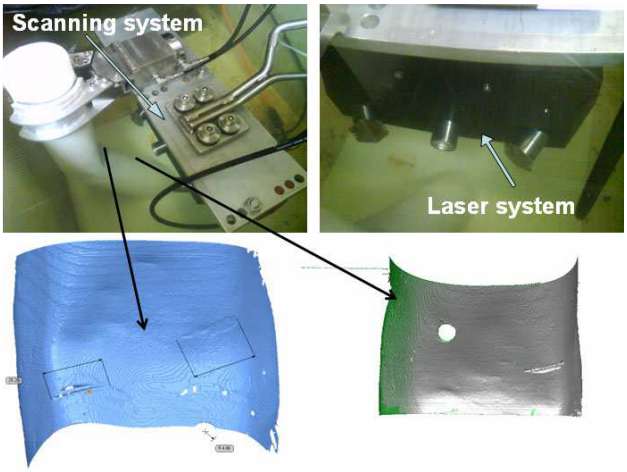

Figure 6: A scanning system and a model reconstructed from its scans [22]. The model clearly reveals that there is a big dent in the pipe and information such as location, area, and volume of the defects can be estimated for repair/replacement.

1.3 Some issues

Range images are usually corrupted by imaging noise, occlusion, holes, spikes and background clutter. Range cameras capture a scene rather than just the object of interest, and therefore the background and other irrelevant objects are also captured. This is especially the case for the range images as shown in Figure 1 which were captured using the Perceptron, ABW and Kinect range cameras. Therefore it may be necessary to segment the data to separate the foreground from the background to facilitate subsequent data processing, e.g. for object modeling and recognition.

One way to achieve this is to make use of the different reflection characteristics from the foreground and the background. Such an idea was adopted in [12] to

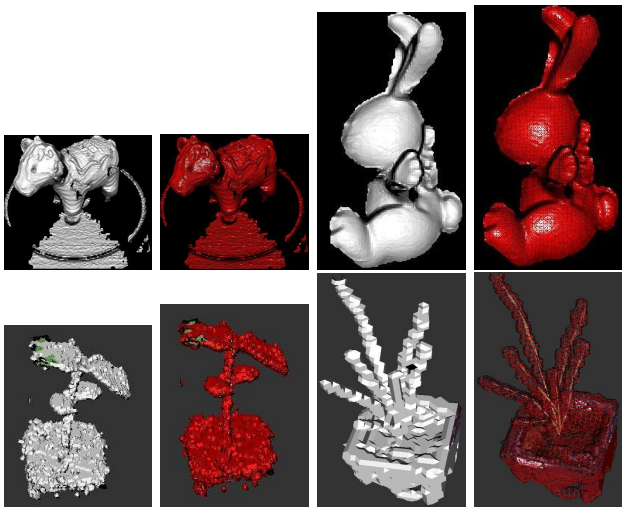

Figure 7: Visualization of real range images (top row) and point clouds (bottom row) as surface (odd column) and triangular mesh wireframe (even column) respectively. For the point clouds, they are superimposed with the original ones with intensity information. Top row: left two: cow42, right two: bunny120. Bottom row: left two: soyabean, right two: wheat.

capture the range images in the fourth row of Figure 1 using a Minolta Vivid 700 range camera. In addition, this camera also outputs the 3D Cartesian points in a text file with 4 blocks of a size of 200 by 200 elements respectively: flag, x coordinate, y coordinate and z coordinate, where $flag(i, j)$ shows whether the point at position (i, j) is valid or not: if $flag(i, j) = 1$, the point $(x(i, j), y(i, j), z(i, j))$ is valid and can be used. Otherwise, the point will not be considered. To help further analysis, these four blocks are rendered for clearer visualization as shown in the bottom row of Figure 7. It can be seen that cow42 was cluttered by the background of the ground, and there are holes in the two eyes, neck, back, and the upper parts of the front and hind legs of cow42. Various spikes in the eyebrows and ears of cow42, and artefacts in the arms of bunny120 are also seen. The holes are mainly caused by lack of data received by the range camera, due to weakly reflective materials or self-occlusion of one part by another part of the object of interest. Spikes and artefacts are mainly caused by the discontinuity of different parts of an object in depth, reflectivity of materials, or surface orientation in the viewing direction.

Cluttered background has been a serious problem for image matching/registration. For instance, it is very difficult to distinguish between the fixed background points from those "moving" points on the foreground objects when a turntable is used for rotating an object for data capture. This is illustrated in the third row of Figure 1 for the images captured by the Kinect sensor. Ideally the features of the background should be stationary and all the points that belong to the background should not be subjected to any transformation. At the same time the points on the foreground objects should be subjected to the transformation due to the movement of the turntable. This shows that the process of data acquisition should be carefully designed to make sure that only the object of interest is captured without other objects or background.

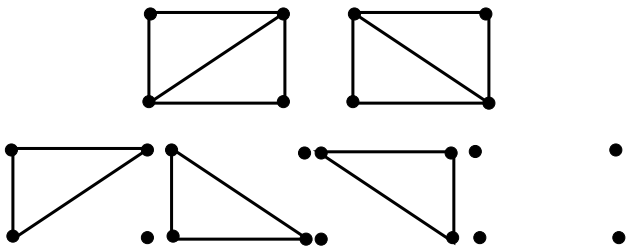

Figure 8: Examples of triangulation of neighboring points in a range image captured by a Minolta Vivid 700 range camera as a mesh for the representation of the surface of the object of interest.

1.4 Notations

The following notations will be used throughout this article. The letters in bold face denote vectors or matrices, lower case letters denote scalars, $|\cdot|$ denotes the absolute value of a scalar or the count of the elements in a set, $||\cdot||$ denotes the Euclidean norm of a vector, $\mathbf{I}$ denotes the identity matrix, $\mathbf{a} \cdot \mathbf{b}$ denotes the dot product of vectors $\mathbf{a}$ and $\mathbf{b}$, $\mathbf{a} \times \mathbf{b}$ denotes the cross product of vectors $\mathbf{a}$ and $\mathbf{b}$, the superscript T denotes a transpose of a vector or a matrix, $\lfloor\rfloor$ denotes the round operation of a real number to a nearest smaller integer, and $det(\mathbf{A})$ denotes the determinant of a matrix $\mathbf{A}$.

While the range images can easily be captured using the commercial off-the-shelf range cameras/sensors/finders, this article also provides some backgrounds to the readers about one of the most challenge technology in range imaging: the issues for the registration of overlapping range images. Image registration has been a critical stage for producing a successful range imaging applications as outlined in Section 1.2 above. To this end, various techniques will be discussed with enough details to allow readers to follow and implement for practical experiments. The remainder of this article is thus organized as follows. After the range images are captured or generated, then they are normally inspected for quality assurance by using open source software such as Meshlab [24] or 3D programming language such as Java3D [25]. Section 2

discusses various visualization methods for the extraction of triangular meshes from given range images or point clouds prior to further analysis. Section 3 outlines the existing SHOT algorithm [26] for feature extractions and matchings (FEM) which are critically important to image registration. Large proportion of false positives can be induced due to lack of features, imaging noise, or cluttered background during the point matching stage, resulting in an inaccurate estimation of the underlying transformation. Inspired by the gentle adaptive boosting [28], the random sample consensus (RANSAC) method [27] has been widely employed for combating false positives [26], Section 4 outlines an alternative algorithm to complement the RANSAC method with an added advantage of easy implementation, and Section 5 presents experimental results using real data captured using a Minolta Vivid 700 range camera and demonstrates whether the typical overlapping range images can be aligned with reasonable accuracy using typical FEM only. Finally, Section 6 draws some conclusions and suggests further readings.

2 Visualization of 3D surface data

After range images have been captured, it is usually necessary to visualize and inspect them for quality, interactive operation, and further analysis. Since range images can be treated as point clouds without taking the neighborhood information into account, two methods for the extraction of triangular meshes from given range images and point clouds will be discussed in the following subsections respectively.

2.1 Range images

While range images can be converted to a Cartesian representation of points (x, y, z) in the 3D space, they usually refer to their Cartesian representation. Range images contain a set of points in a certain order and in which overlap of one part against another does not occur. In this case, a triangular mesh can be easily extracted from a given range image for the visualization of the surface of the object of interest that it represents (Figure 8) where neighboring points can be easily identified. For a valid point $(x(i,j), y(i,j), z(i,j))$ in a range image captured by a Minolta Vivid 700 range camera, its three neighbors are $(x(i+1,j), y(i+1,j), z(i+1,j))$, $(x(i,j+1), y(i,j+1), z(i,j+1))$, and $(x(i+1,j+1), y(i+1,j+1), z(i+1,j+1))$. The number of triangles that can be established is dependent on the distribution of the neighboring points. For a valid top-left point $(x(i,j), y(i,j), z(i,j))$, its 3 neighbors in the right and underneath can be considered: if all of them are valid, then two triangles can be generated with two different configurations (top row), normally the one with a short diagonal length is preferred, so that the created triangles become as equilateral as possible; if two of them are valid (bottom left three), then a unique triangle will be generated. Otherwise, no triangle (bottom right) will be generated. This process can be repeated until all the points have been visited. The renderings of the cow42 and bunny120 images were implemented in Java3D and are presented in Figure 7 as the solid surface and triangular mesh wireframe respectively.

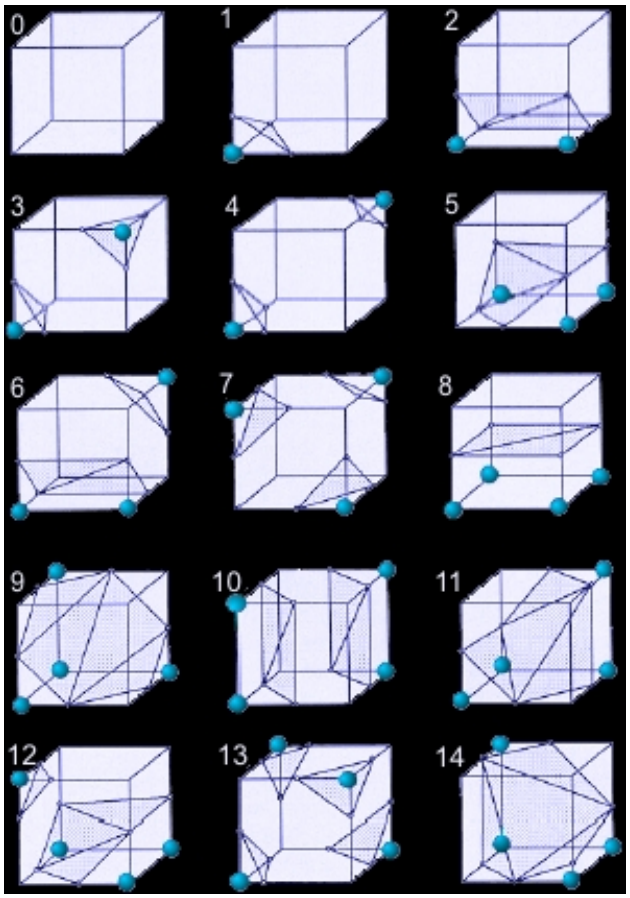

Figure 9: Different configurations between the states of the vertices of a cube and the given surface represented by the given point cloud, leading to the generation of at most 5 triangles [30].

2.2 Point clouds

The points in a point cloud do not have any order and some points may occlude each other, depending on the direction from which the point cloud is viewed. In this case, a neighbor is not well defined in the sense

of distance or the structure of the object of interest. In order to identify a neighbor nearest to a point, all the other points may have to be visited once, unless some special data structures such as a k-D tree [29] have already been built in advance for the storage of the points. For large point clouds with hundreds of thousands of points, or even millions of points, it is time consuming to identify even just a small number of nearest neighbors of a point.

In order to extract a triangular mesh from a point cloud for visualization and analysis, the classical method, marching cubes [30], can be applied. It includes three main steps: (i) Locate the surface corresponding to a user-specified value. Suppose that the given point cloud contains N points: $\mathbf{p}_i = (x_i, y_i, z_i)^T$ $(i = 1, 2, \cdots, n)$, the spans or bounding box of these points along different axes need to be firstly found out: minimum and maximum x_{min}, x_{max}, y_{min}, y_{max}, z_{min}, and z_{max} along the x, y and z axes, respectively: $x_{min} = \min_i x_i$, $x_{max} = \max_i x_i$, $y_{min} = \min_i y_i$, $y_{max} = \max_i y_i$, $z_{min} = \min_i z_i$, and $z_{max} = \max_i z_i$. Then it is necessary to determine the dimensionality of the bounding box: n_x, n_y and n_z, which leads to the determination of the cube size: width s_x, height s_y and depth s_z as $s_x = (x_{max} - x_{min})/n_x$, $s_y = (y_{max} - y_{min})/n_y$ and $s_z = (z_{max} - z_{min})/n_z$. Finally, whether a voxel (u, v, w) contains at least a point can be determined. If so, it can be represented as $S(u, v, w) = 1$; if not, it can be represented as $S(u, v, w) = 0$. All the points $\mathbf{p}_i = (x_i, y_i, z_i)^T$ of the surface can be voxelized as: if $u = \lfloor (x_i - x_{min})/s_x \rfloor$, $v = \lfloor (y_i - y_{min})/s_y \rfloor$, and $w = \lfloor (z_i - z_{min})/s_z \rfloor$, then $S(u, v, w) = 1$. Based on such information whether a voxel is occupied, it is possible to determine whether a cube vertex is inside and outside the surface represented by the given point cloud;

(ii) Create triangles. Interestingly, for each cube, it has 8 vertices and each has two possible states: inside or outside the surface. Thus, altogether, there are 256 different configurations between the states of the vertices of the cube and the surface, which can be enumerated and defined as a look-up table for the sake of computational efficiency. Based on the symmetry, these 256 cases can be further reduced to 15 as illustrated in Figure 9. Based on such configurations, a mask can be created to index which edge has been intersected, from which the intersection point can be linearly interpolated based on the *isovalue* between 0 and 1. Suppose the edge has two vertices $\mathbf{V}_1$ with a value of 0 and $\mathbf{V}_2$ with a value of 1, then the intersection point is $\mathbf{V} = \mathbf{V}_1 + isovalue(\mathbf{V}_2 - \mathbf{V}_1)$. Based on the configurations of the vertices and these interpolated points, a number of, at most five, triangles can be generated; and

(iii) Calculate the normals to the shape at each vertex. For each vertex of a triangle, the neighbouring triangles $\{\mathbf{T}_i\}$ that share this vertex can be identified. The normal vector $\mathbf{n}_i$ and area a_i of each triangle $\mathbf{T}_i$ with vertices $\mathbf{V}_0$, $\mathbf{V}_1$, and $\mathbf{V}_2$ can be calculated as: $\mathbf{n}_i = (\mathbf{V}_1 - \mathbf{V}_0) \times (\mathbf{V}_2 - \mathbf{V}_0)/||\mathbf{V}_1 - \mathbf{V}_0||||\mathbf{V}_2 - \mathbf{V}_0||$ and $a_i = 0.5|(\mathbf{V}_1 - \mathbf{V}_0) \times (\mathbf{V}_2 - \mathbf{V}_0)|$. For $\mathbf{n}_i$, there is an ambiguity about its sign. Considering the captured surface is visible from the scanner, then the following rule can be used to disambiguate its sign: if $\mathbf{n}_i \cdot (\mathbf{V}_0 + \mathbf{V}_1 + \mathbf{V}_2)/3 > 0$, then $\mathbf{n}_i \leftarrow -\mathbf{n}_i$. Finally, the normal vector $\mathbf{n}$ at vertex $\mathbf{v}$ is calculated as a weighted average of $\mathbf{n}_i$ with weights defined by a_i: $\mathbf{n} = \sum_{\mathbf{T}_i} a_i \mathbf{n}_i / \sum_{\mathbf{T}_i} a_i$ and $\mathbf{n} \leftarrow \mathbf{n}/||\mathbf{n}||$. Such scheme shows that the normal vectors of large triangles will dominate and thus is likely to prevent the normal vectors of small and oblique triangles from distorting that of the vertex significantly.

Figure 7 shows some examples for the visualization of the soyabean and wheat point clouds stored in PLY format [15] as rendered solid surface and triangular mesh wireframe in Java3D, superimposed with the original ones with intensity information respectively, where $isovalue = 0.0025$, $n_x = 64$, $n_y = 64$, and $n_z = 64$. By comparison, it can be seen that the range images are better rendered with neighborhood information and smooth surface than the point clouds with the block effect. This is because it is difficult to decide the dimensionality of the bounding box. If it is too small, then the rendered point cloud will lose details due to low resolution. If it is too large, then it is likely to create holes and discontinuity due to variation of density and lack of points in certain voxels. On the other hand, with the dimensionality of the bounding box increasing, the memory required will increase dramatically in the order of $O(n_x n_y n_z)$.

3 Point match establishment

The registration of overlapping range images has been one of the most basic tasks to initiate 3D imaging analysis and applications for structural similarity and transformation. For still targets, the distance between any two points in all range images is assumed constant. In this case, the underlying transformation that brings one range image R_1 into the best possible alignment with another R_2 can be represented by a rigid rotation matrix $\mathbf{R}$, with $\mathbf{R}^T\mathbf{R} = \mathbf{I}$ and the determinant of $\mathbf{R}$ being equal to 1: $det(\mathbf{R}) = 1$, and a translation vector $\mathbf{t}$. $(\mathbf{R}, \mathbf{t})$ represent the relative orientation and position of the two viewpoints from which the two given range images R_1 and R_2 were captured.

In this section, the principle of the existing Signature of Histograms of OrienTations (SHOT) method [26] is outlined due to its good performance for registra-

tion of range images [31]. The SHOT method is applicable to both range images and point clouds. The neighborhood of a point in a given range image can be easily identified as discussed in the last section. In the case of a given point cloud, the neighborhood of a point can be defined as those within a certain distance from the point of interest or from the triangular mesh generated by a chosen method such as marching cubes as outlined in the last section. The main idea of the SHOT method is to encode the local surface orientation variation as a feature of a point on the surface. The features of the points can be matched for point correspondences, from which the underlying transformation rotation matrix $\hat{\mathbf{R}}$ and translation vector $\hat{\mathbf{t}}$ can be estimated in the least squares sense using the quaternion method [32] with a closed form solution, for example. A correspondence is such a pair of points $(\mathbf{p}, \mathbf{p}')$ from two overlapping range images that represent the same point on the surface of the object of interest.

The SHOT method will be outlined in such a way as a complete solution to 3D surface registration, instead of just as a 3D surface descriptor. In this case, the outline will include the following steps: feature point detection, description, matching and underlying transformation estimation in the following subsections.

3.1 Motivation

For feature extraction and matching (FEM), it is normally assumed that the points in the range images can be represented using some features f that are invariant to the coordinate system in which they are represented and the correspondences $(\mathbf{p}, \mathbf{p}')$ would have almost the same values in these features: $f(\mathbf{p}) = f(\mathbf{p}')$, due to imaging sampling and noise. This is usually a necessary condition for a pair of points that are corresponding to each other, but in most cases it is not sufficiently robust enough. Henceforth a large proportion of false correspondences is found in practice. The false positive rate is sensitive to how the invariant features f are defined. There are also a number of influencing factors such as imaging noise, image resolution, expressiveness of the features to be extracted, and the geometry of the object of interest.

3.2 Feature point selection

While the SHOT algorithm did not include a feature point selection step but randomly selected points as input, for the sake of computational efficiency and establishing more reliable point matches, we outline an effective method proposed in [33] for the task. The point matches are more likely to be correct when they are locally distinct. These points are usually called feature points, key points, or interest points. They are usually the points with extreme values in certain features of interest over their neighborhood. To reduce the sensitivity of feature points to the object geometry, imaging noise and resolution, the neighborhood should be large enough and the estimated features should be relatively stable to these disturbing factors. The larger the neighborhood, the fewer the feature points to be selected and vice versa.

Given a range image and the Cartesian coordinates of points $\mathbf{p}_i$, a triangular mesh can be generated through triangulating these points as discussed in the last section. Then the method in [34] can be applied for the estimation of the shape indexes $p_s(i)$ and surface types $p_t(i)$ of these points through the parameters such as the normal vectors $\mathbf{n}_i$, principal curvatures, Gaussian curvatures and mean curvatures of these points.

All the neighboring points of a point $\mathbf{p}_i$ are identified as those whose distances are smaller than a threshold τ_1: $N_i = \{\mathbf{p}_{ik} | \, ||\mathbf{p}_{ik} - \mathbf{p}_i|| < \tau_1\}$ where τ_1 is a threshold set as $\tau_1 = 2r$, and r is the average of the distances of the nearest points (DNPs) in the range image. Then the minimum, average s_m and maximum of the shape indexes of these points are identified and calculated in N_i. If $p_s(i)$ is a minimum and smaller than a threshold $(1 - \beta)s_m$ or $p_s(i)$ is a maximum and larger than a threshold $(1 + \alpha)s_m$, then point $\mathbf{p}_i$ is selected as a feature point, where $\beta = 0.02$ and $\alpha = 0.05$. The larger the parameters τ_1, α and β are, the fewer the feature points will be detected.

3.3 Local reference frame estimation

In order to describe the feature points for matching, some invariants that do not change from one range image to another have to be extracted in the local reference frame (LRF) [26]. A weighted scatter matrix method is proposed for the task over the neighboring points of the selected feature points $\mathbf{p}_i$: $N_i = \{\mathbf{p}_{ik} | \, ||\mathbf{p}_{ik} - \mathbf{p}_i|| < \tau_2\}$ where τ_2 is a threshold set as $\tau_2 = 15r$ and must be large enough to contain enough neighboring points for the characterization of the local geometry. The weighted scatter matrix $\mathbf{C}$ is constructed as: $\mathbf{C} = \sum_k (\tau_2 - ||\mathbf{p}_{ik} - \mathbf{p}_i||)(\mathbf{p}_{ik} - \mathbf{p}_i)(\mathbf{p}_{ik} - \mathbf{p}_i)^T$, where the weight of each neighboring point $\mathbf{p}_{ik}$ is defined as the difference between τ_2 and its distance $||\mathbf{p}_{ik} - \mathbf{p}_i||$ to the feature point $\mathbf{p}_i$. While $\mathbf{C}$ is a symmetric matrix, then the Jacobi method can be used to estimate its eigenvalues λ_1, λ_2, and λ_3 in descending order, and their corresponding eigenvectors $\mathbf{v}_1$, $\mathbf{v}_2$, and $\mathbf{v}_3$. Such vectors have an ambiguity of sign. To construct a unique LRF, these signs should be disambiguated. The numbers $n_1' = |N_i^1|$ and $n_2' = |N_i^2|$ of points in N_i on the positive and negative sides of $\mathbf{v}_1$ can be counted as: $N_i^1 = \{\mathbf{p}_{ik} | (\mathbf{p}_{ik} - \mathbf{p}_i) \cdot \mathbf{v}_1 > 0\}$

and $N_i^2 = \{\mathbf{p}_{ik} | (\mathbf{p}_{ik} - \mathbf{p}_i) \cdot \mathbf{v}_1 < 0\}$. Suppose that more points are always on the positive side, then if $n_1' < |N_i| - n_1'$, then the direction of $\mathbf{v}_1$ should be flipped as: $\mathbf{v}_1 \leftarrow -\mathbf{v}_1$. The same method is applied to $\mathbf{v}_3$ to disambiguate its sign. Finally $\mathbf{v}_2 = \mathbf{v}_3 \times \mathbf{v}_1$.

3.4 Feature point description

To describe the distribution of neighboring points, a bounding sphere with a radius of τ_2 centred at the feature point $\mathbf{p}_i$ is split into sections (Figure 10): $n_a = 8$ in azimuth, $n_e = 2$ in elevation and $n_r = 2$ in radius. A local histogram with $n_h = 10$ bins is built to describe how the neighboring points in each section vary in normal vector relative to that of the feature point $\mathbf{p}_i$. All the local histograms from different sections are finally concatenated as a $n_a \times n_e \times n_r \times n_h = 8 \times 2 \times 2 \times 10 = 320$ dimensional vector $\mathbf{f}_i$ for the representation of each feature point $\mathbf{p}_i$. For the sake of robustness to imaging noise and resolution, such features $\mathbf{f}_i$ are usually normalized as: $\mathbf{f}_i \leftarrow \mathbf{f}_i / ||\mathbf{f}_i||$.

For a feature point $\mathbf{p}_i$, its feature vector $\mathbf{f}_i$ is initialized as a zero vector with all components being zero: $\mathbf{f}_i = \mathbf{0}$. For each neighbouring point $\mathbf{p}_{ik}$, which section it lies within can be identified: $fx = (\mathbf{p}_{ik} - \mathbf{p}_i) \cdot \mathbf{v}_1$, $fy = (\mathbf{p}_{ik} - \mathbf{p}_i) \cdot \mathbf{v}_2$, $fz = (\mathbf{p}_{ik} - \mathbf{p}_i) \cdot \mathbf{v}_3$, $fd = ||\mathbf{p}_{ik} - \mathbf{p}_i||$, the azimuth index is estimated as: $a_k = \lfloor atan2(fy, fx)/da \rfloor$, the elevation index is estimated as: $e_k = \lfloor acos(z/fd)/de \rfloor$, the radius index is estimated as: $r_k = \lfloor fd/dr \rfloor$, the bin index in the local histogram is estimated as: $b_k = [(1 + c_k)n_h/2]$, and a concatenated histogram $\mathbf{f}_i$ is built as $f_i[((a_k n_e + e_k) * n_r + r_k) * n_h + b_k] \leftarrow f_i[((a_k n_e + e_k) * n_r + r_k) * n_h + b_k] + 1$ where $da = 2\pi/n_a$, $de = \pi/n_e$, $dr = \tau_2/n_r$, and c_k is the cosine of the including angle between its normal vector $\mathbf{n}_{ik}$ and the normal vector $\mathbf{n}_i$ at the central point $\mathbf{p}_i$: $c_k = \mathbf{n}_{ik} \cdot \mathbf{n}_i$. Note that the vote of 1 from a neighboring point may be shared by neighboring bins/sections, dependent on how far away the point is from the center of neighboring bins/sections. This is an optional operation and its implementation details are thus omitted in this article. In some cases, such histogram interpolation may lead to slightly more accurate representation of the feature points.

3.5 Feature point matching

Given two range images R_1 and R_2 and their Cartesian representation of point sets $\mathbf{P}$ and $\mathbf{P}'$, the above steps can be used to extract two sets of feature points $\mathbf{p}_i$ $(i = 1, 2, \cdots, n_1)$ and $\mathbf{p}_j'$ $(j = 1, 2, \cdots, n_2)$ and their descriptors $\mathbf{f}_i$ and $\mathbf{f}_j'$ respectively. For each point $\mathbf{p}_i$, the difference d_{ij} between its descriptor $\mathbf{f}_i$ and any feature descriptor $\mathbf{f}_j'$ can be calculated as $d_{ij} = ||\mathbf{f}_i - \mathbf{f}_j'||$. Those points $\mathbf{p}_i'$ and $\mathbf{p}_i''$ can be identified with the minimum and second minimum differences $d_{ii'}$ and $d_{ii''}$: if $\mathbf{p}_i$ and $\mathbf{p}_i'$ have the same surface type and $d_{ii'} < \gamma d_{ii''}$, then $(\mathbf{p}_i, \mathbf{p}_i')$ is established as a point match where $\gamma = 0.975^2$ and controls how different $\mathbf{p}_i'$ and $\mathbf{p}_i''$ should be when a putative point match (PPM) $(\mathbf{p}_i, \mathbf{p}_i')$ can be established. It helps to remove ambiguity in establishing the correct point matches. The larger the value of γ, the more PPMs there would be established, and vice versa.

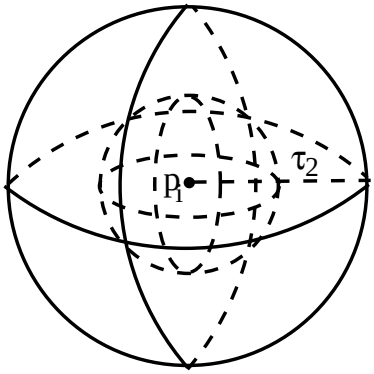

Figure 10: The neighboring sphere with a radius of τ_2 of feature point $\mathbf{p}_i$ has been split into 4 sections in azimuth, 2 sections in elevation, and 2 sections in radius for the illustration of the SHOT feature description.

3.6 Underlying transformation estimation

Suppose $(\mathbf{p}_i, \mathbf{p}_i')$ $(i = 1, 2, \cdots, N)$ is a set of PPMs established in the last subsection, then the underlying transformation $(\mathbf{R}, \mathbf{t})$ can be estimated from the following objective function that minimises the sum of the squares of the errors $e_i = ||\mathbf{p}_i' - \mathbf{R}\mathbf{p}_i - \mathbf{t}||$ of these point matches:

$$J(\mathbf{R}, \mathbf{t}) = \min_{\mathbf{R}, \mathbf{t}} \sum_i e_i^2. \tag{2}$$

The quaternion method [32] was used to optimize this objective function with a closed form solution in the least squares sense. In this objective function, all the PPMs are equally treated, no matter how large their errors e_i are. Such objective function normally will not lead to an accurate estimation of the underlying transformation $(\mathbf{R}, \mathbf{t})$, since the false matched points may dominate this objective function. To get an accurate estimation of $(\mathbf{R}, \mathbf{t})$, the correct and false matched points must be distinguished from each other and the estimation should be based on the correct ones only.

3.7 Computational complexity

The SHOT method has a computational complexity of $O(n)$ for the estimation of the shape indexes and surface types of points of interest and $O(n^2)$ for calculating the inter-point distances for the search of neighbouring points for the detection and description of the feature points: n is the number of points in the given range image, $O(n_3 n_4)$ for the estimation of the local reference frame where $n_3 \leq n$ is the number of feature points detected and $n_4 \leq n$ is the number of neighbouring points of a feature point, $O(n_3 n_4)$ for the building of the feature descriptors, $O(n_3)$ for the normalization of the feature descriptors, $O(n_3^2)$ for the establishment of the PPMs and $O(N)$ for the estimation of the underlying transformation where $N \leq n_3$ is the number of the finally established PPMs. Thus, it has an overall computational complexity of $O(n^2)$. Without using the k-D tree [29], for example, to speed up the search for the nearest neighbors, the computation in $O(n^2)$ will be extremely time consuming especially when n is larger than hundreds of thousands.

4 Point match evaluation

Suppose $(\mathbf{p}_i, \mathbf{p}'_i)$ $(i = 1, 2, \cdots, N)$ is a set of PPMs established using the SHOT algorithm outlined in the last section between two given overlapping range images. Due to various factors such as the imaging noise, occlusion, cluttered background and relative simple geometry that the object surface has, the established point matches usually include a large proportion of false positives. Such false positives would lead to inaccurate estimations of the underlying transformation and thus inaccurate alignment of the given range images. While the random sample consensus (RANSAC) method [27] has been widely used to combat such false positives [26], it has a difficulty in classifying the PPMs into inliers and outliers and measuring the quality of the candidate underlying transformations. In this section, a novel alternative method is proposed to advance the RANSAC method with better performance. It assesses the reliabilities of these point matches as weights in the unit interval that would lead the underlying transformation to be estimated in the weighted least squares (WLS) sense. Actually, the proposed method is applicable to the PPMs established by any typical FEM method between two partially overlapping range images subject to rigid transformations and it has an advantage of easy implementation.

4.1 Motivation and ideas

The proposed method was inspired by the gentle adaptive boosting [28] for the classification tasks. Given a set of training examples, it includes three main steps: (i) Train a set of weak classifiers; (ii) Estimate the weighted average error and thus the boosting parameter so that the weights of correctly classified examples will be decreased, and those of the incorrectly classified examples will be increased; and (iii) Update the weights of all the training examples. These steps are repeated until either the maximum number of iterations has been reached or the average error of the weak classifier is too large. Finally a decision function is built for the classification of a given example. The sign of the weighted average of the predictions of all the weak classifiers indicates the class that it belongs to: where the larger the error the weak classifier produces, the smaller the weight it contributes.

The problem for the evaluation of the established PPMs is different from the classification problem above in two aspects: (i) no training examples are available whether they are correct or incorrect with a label of 1 or -1, (ii) it is essentially a regression problem, rather than a classification one. Thus, we have to adapt it. The main idea is to penalize such PPMs with large errors from the weighted average. To make sure that the method is robust to various range images for the representation of different objects subject to different transformations, such errors are normalized by their standard deviation. Then the weights of different PPMs are updated with a closed form solution. These steps are repeated until either the maximum number of iterations has been reached or the weighted average of the errors of the PPMs is small enough. These steps are detailed in the following subsections.

4.2 Main steps

After a set of PPMs $(\mathbf{p}_i, \mathbf{p}'_i)$ is established, a real number w_i in the unit interval [0, 1] to represent their reliabilities as weights is introduced. The larger the weight, the more likely we believe that the PPM $(\mathbf{p}_i, \mathbf{p}'_i)$ is correct. w_i was firstly initialized as $w_i = 1/N$ due to the lack of knowledge about their true reliabilities.

Then w_i was normalized as: $w_i \leftarrow w_i / \sum_j w_j$. The underlying transformation rotation matrix $\mathbf{R}$ and translation vector $\mathbf{t}$ can be estimated in the WLS sense from the following objective function that minimizes the weighted average of the squared errors e_i^2 of all the PPMs $(\mathbf{p}_i, \mathbf{p}'_i)$:

$$J(\mathbf{R}, \mathbf{t}) = \min_{\mathbf{R},\mathbf{t}} \sum_i w_i e_i^2. \tag{3}$$

The quaternion method [32] was also used to optimize

this objective function with a closed form solution. Then the weighted average μ and standard deviation σ of the errors e_i of different PPMs $(\mathbf{p}_i, \mathbf{p}'_i)$ can be computed subsequently as:

$$\mu = \sum_i w_i e_i, \sigma = \sqrt{\sum_i w_i (e_i - \mu)^2}. \quad (4)$$

To update the weight w_i and make sure that the method is robust to the variation of imaging resolution, imaging noise and magnitude of the underlying transformation, firstly, we normalize the errors e_i of the PPMs $(\mathbf{p}_i, \mathbf{p}'_i)$ from μ over σ as:

$$\hat{e}_i = (e_i - \mu)/\sigma, \quad (5)$$

then we design the following objective function, which minimizes the weighted average of the squared normalized errors $\hat{e}_i^2$ of the PPMs $(\mathbf{p}_i, \mathbf{p}'_i)$ regularized by the Shannon entropy of these weights in the framework of entropy maximization [36]:

$$J(\mathbf{W}) = \min_{\mathbf{W}} \sum_i w_i \hat{e}_i^2 + \frac{1}{\beta} \sum_i w_i (\log w_i - 1) \quad (6)$$

where $\mathbf{W} = \{w_1, w_2, \cdots, w_N\}$, and parameter β balances the contribution of the two terms in the function and is set to $\beta = 4$ in this article. Setting the first order derivative of this objective function to zero leads to: $w_i = exp(-\beta \hat{e}_i^2)$.

The above process can be repeated until either the maximum number M of iterations has been reached or μ is smaller than the average s of the DNPs in the given images. In order to learn from different iterations, we select the larger of the existing $w_i^{(k)}$ and the newly estimated one $w_i^{(k+1)}$ at each iteration k as the final updated weighted value:

$$w_i^{(k+1)} = \max(w_i^{(k+1)}, w_i^{(k)}). \quad (7)$$

4.3 Summary of the proposed method

To summarize all the components and innovations described in the previous section, we have the following algorithm for the automatic registration of two overlapping range images R_1 and R_2:

1: Use a typical FEM method to select a set of feature points from each range image
2: Use a typical FEM method to establish a set of PPMs $(\mathbf{p}_i, \mathbf{p}'_i)(i = 1, 2, \cdots, N)$ between R_1 and R_2
3: Initialize the weights w_i of the PPMs $(\mathbf{p}_i, \mathbf{p}'_i)$ as $w_i = 1/N$, iteration index $k = 0$, the maximum number M of iterations, and the average s of the DNPs in R_1 and R_2
4: **do**
5: $k \leftarrow k + 1$
6: Use Equation 3 to estimate ($\mathbf{R}$, $\mathbf{t}$)
7: Use Equation 4 to compute μ and σ
8: Use Equation 5 to normalise e_i
9: Use Equation 7 to update w_i
10: **while** $k < M$ and $\mu > s$

Our method is based on the normalized error penalization and thus is called NEP. It can be seen that it has a computational complexity of $O(N)$ for estimating ($\mathbf{R}$, $\mathbf{t}$), $O(N)$ for calculating μ and σ, $O(N)$ for normalizing e_i, and $O(N)$ for updating w_i. Thus, it has an overall linear computational complexity of $O(N)$ in the number N of the established PPMs.

5 Experimental results

This section briefly outlines the experimental validation of the SHOT method and the proposed NEP algorithm compared with the widely used RANSAC method [27] for FEM. The experiment aims at providing some evidence of examining the established point matches, the effectiveness of the FEM as given by the SHOT/RANSAC/NEP methods and to assess their limitations. The performance of these techniques is assessed through the estimated transformation rotation matrix $\hat{\mathbf{R}}$ and translation vector $\hat{\mathbf{t}}$ of the point matches $(\mathbf{p}_i, \mathbf{p}'_i)$ and to compare with that of the ground truth. The estimated $(\hat{\mathbf{R}}, \hat{\mathbf{t}})$ can also be refined using the SoftICP [35] method which is a variant of the traditional iterative closest point (ICP) algorithm [32]. The refined $(\hat{\mathbf{R}}, \hat{\mathbf{t}})$ can sometimes be used as ground truth. The difference of $(\hat{\mathbf{R}}, \hat{\mathbf{t}})$ before and after ICP refinement can give some indication of the effectiveness of the FEM methods [37]. Such a strategy is useful especially when the ground truth $(\mathbf{R}, \mathbf{t})$ is either partially known or unavailable, as it is the case for the data being used in this article.

Of particular interests are the relative errors such as the $e_{\mathbf{h}}$, e_θ, and $e_{\mathbf{t}}$ of the estimated rotation axis $\hat{\mathbf{h}}$, rotation angle $\hat{\theta}$, and translation vector $\hat{\mathbf{t}}$ with respect to the ground truth. The error is expressed in percentage difference between the estimated and reference rotation axes $\hat{\mathbf{h}}$ and $\mathbf{h}$, rotation angles $\hat{\theta}$ and θ, and translation vectors $\hat{\mathbf{t}}$ and $\mathbf{t}$ of the underlying transformation: $e_{\mathbf{h}} = ||\hat{\mathbf{h}} - \mathbf{h}|| \times 100\%$, $e_\theta = (\hat{\theta} - \theta)/\theta \times 100\%$, and $e_{\mathbf{t}} = ||\hat{\mathbf{t}} - \mathbf{t}||/||\mathbf{t}|| \times 100\%$ where the rotation angle $\hat{\theta}$ and rotation axis $\hat{\mathbf{h}}$ are estimated from the estimated rotation matrix $\hat{\mathbf{R}}$ as: $\hat{\theta} = \frac{180}{\pi} acos((r_{11} + r_{22} + r_{33} - 1)/2)$, $\hat{\mathbf{h}} \leftarrow \hat{\mathbf{h}}/||\hat{\mathbf{h}}||$, $\hat{\mathbf{h}} = (\frac{r_{32} - r_{23}}{\sin\hat{\theta}}, \frac{r_{13} - r_{31}}{\sin\hat{\theta}}, \frac{r_{21} - r_{12}}{\sin\hat{\theta}})^T$, and $\hat{\mathbf{R}} = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix}$. Other assessment param-

Table 1: The estimated rotation matrix $\hat{\mathbf{R}}$, rotation axis $\hat{\mathbf{h}}$, rotation angle $\hat{\theta}$, and translation vector $\hat{\mathbf{t}}$, and computational *time* in seconds of different methods over different pairs of overlapping range images.

Image pair	Method	$\hat{\mathbf{R}}$			$\hat{\mathbf{h}}$	$\hat{\theta}$	$\hat{\mathbf{t}}$	time(sec)
cow42-37	SHOT	0.99	0.01	0.05	-0.99	24.75	59.58	
		-0.03	0.91	0.41	0.11		564.75	
		-0.04	-0.41	0.91	-0.06		-120.00	
	SHOT+SoftICP	0.55	-0.77	-0.32	0.23	57.88	-438.86	499
		0.66	0.64	-0.38	-0.49		-459.92	
		0.50	0.00	-0.86	0.84		-181.78	
	SHOT+RANSAC	0.65	0.55	-0.52	0.09	49.79	-697.63	
		-0.59	0.80	0.11	-0.66		142.06	
		0.48	0.24	0.84	-0.75		-201.92	
	SHOT+RANSAC+SoftICP	0.64	0.54	-0.54	0.02	50.31	-723.69	186
		-0.56	0.81	-0.16	-0.70		215.17	
		0.53	0.20	0.82	-0.72		-233.44	
	SHOT+NEP	0.64	0.56	-0.52	-0.00	50.14	-695.66	
		-0.56	0.81	0.18	-0.68		245.91	
		0.53	0.17	0.83	-0.73		-220.30	
	SHOT+NEP+SoftICP	0.64	0.54	-0.54	0.02	50.31	-723.66	206
		-0.56	0.81	0.16	-0.70		215.17	
		0.53	0.20	0.82	-0.72		-233.43	
bunny120-60	SHOT	0.55	0.34	-0.76	0.11	56.64	-503.54	
		-0.43	0.90	0.09	-0.88		57.09	
		0.71	0.28	0.65	-0.46		-231.61	
	SHOT+SoftICP	0.51	0.43	-0.74	-0.02	59.28	-499.88	26
		-0.41	0.88	0.23	-0.87		150.88	
		0.76	0.19	0.63	-0.48		-246.60	
	SHOT+RANSAC	0.46	0.60	-0.64	-0.07	62.44	-430.61	
		-0.54	0.77	0.33	-0.76		217.06	
		0.70	0.20	0.69	-0.65		-205.40	
	SHOT+RANSAC+SoftICP	0.51	0.43	-0.74	-0.02	59.28	-499.89	16
		-0.41	0.88	0.23	-0.87		150.86	
		0.76	0.19	0.63	-0.48		-246.60	
	SHOT+NEP	0.55	0.47	-0.69	-0.02	56.84	-464.66	
		-0.45	0.86	0.22	-0.83		146.68	
		0.70	0.19	0.68	-0.55		-207.61	
	SHOT+NEP+SoftICP	0.51	0.43	-0.74	-0.02	59.28	-499.96	25
		-0.41	0.88	0.23	-0.87		150.91	
		0.76	0.19	0.63	-0.48		-246.69	

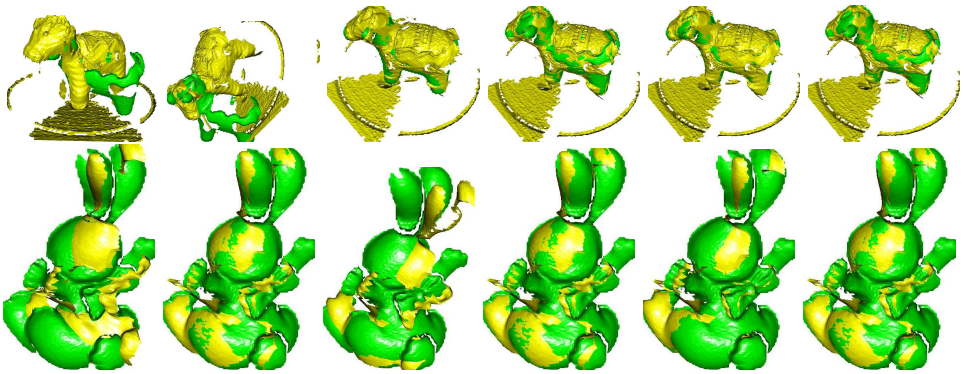

Figure 11: Different overlapping range images are brought into alignment by the original and evaluated PPMs using different evaluation and refinement methods. From left to right column: SHOT, SHOT+SoftICP, SHOT+RANSAC, SHOT+RANSAC+SoftICP, SHOT+NEP and SHOT+NEP+SoftICP. Top row: cow42-37; Bottom: bunny120-60.

Table 2: The relative errors $e_{\mathbf{h}}$, e_θ, and $e_{\mathbf{t}}$ in percentage of the estimated rotation axis $\hat{\mathbf{h}}$, rotation angle $\hat{\theta}$, and translation vector $\hat{\mathbf{t}}$, the average e_μ and standard deviation e_σ of the errors in millimeters of the RCs, and the percentage c_p of correct PPMs of different methods over different pairs of overlapping range images.

Image pair	Method	$e_{\mathbf{h}}(\%)$	$e_\theta(\%)$	$e_{\mathbf{t}}(\%)$	$e_\mu(mm)$	$e_\sigma(mm)$	$c_p(\%)$
cow42-37	SHOT	163.02	-57.24	172.59	1.00	1.29	0.22
	SHOT+RANSAC	8.36	-1.03	10.61	0.74	0.42	5.51
	SHOT+NEP	3.69	-0.00	5.52	0.74	0.43	5.51
bunny120-60	SHOT	14.14	-4.44	16.46	0.25	0.16	7.08
	SHOT+RANSAC	20.58	5.34	18.06	0.26	0.16	7.08
	SHOT+NEP	7.80	-4.12	9.15	0.25	0.16	7.08

eters are: computational time in seconds for the point match establishment and evaluation, the average e_μ and standard deviation e_σ of the registration errors e_i of the reciprocal correspondences (RCs) $(\mathbf{p}_i, \mathbf{p}'_i)$ $(i = 1, 2, \cdots, n_5)$: $e_i = ||\mathbf{p}'_i - \hat{\mathbf{R}}\mathbf{p}_i - \hat{\mathbf{t}}||$ over all the points in the two overlapping range images, $e_\mu = \frac{1}{n_5}\sum_i e_i$ and $e_\sigma = \sqrt{\frac{1}{n_5}\sum_i (e_i - e_\mu)^2}$ and the percentage c_p of correct matches: $c_p = N_c/N \times 100\%$, where $\hat{\mathbf{R}}\mathbf{p}_i + \hat{\mathbf{t}}$ is the closest to $\mathbf{p}'_i$ over all points in the reference range image R_2, $\hat{\mathbf{R}}^T(\mathbf{p}'_i - \mathbf{t})$ is the closest to $\mathbf{p}_i$ over all points in the data range image R_1, and N_c is the number of correct matches with errors smaller than $4r$ with respect to the estimated ground truth $(\mathbf{R}, \mathbf{t})$ from the SoftICP algorithm.

Two pairs of real range images: cow42-37 and bunny120-60 without backgrounds as shown in Figure 1 have been selected for the experiments. The targets are free form surfaces of objects such as cow and bunny and it is assumed that all range images partially overlap and also that the objects exhibit relatively complicated geometry in some places. Otherwise, there is no other assumption for the image matching and registration. All algorithms were implemented and run on a PC with an Intel Xeon E5620 processor with C programming language inside the Microsoft visual studio 2013 (without code optimization).

5.1 Point match establishment

The experimental results are presented in Figure 11 and Tables 1 and 2. In the figure, the yellow pixels represent the transformed range images R_1, and green represents the reference range images R_2. It can be seen from Figure 11 that despite the fact that the point matches were searched from all possible candidates and the percentage of the correct matches is as low as 7%, the estimated underlying transformation is still fairly accurate. This is especially the case for the bunny120-60 images. A relatively accurate underlying transformation provides a good initialization for the SoftICP algorithm which converges quickly within just 25 seconds. However, the cow object consists of background cluttered range images (e.g. cow42) and therefore the point matches between cow42 and other range images, such as the uncluttered cow37, exhibit a large number of false positive matched points. This leads to an inaccurate estimation of the underlying transformation in the least squares sense. Such inaccuracy in the underlying transformation also degrades the refinement by using the SoftICP algorithm.

The parameters of interest for the FEM in both objects are tabulated in Tables 1 and 2. It is seen that the inaccurate transformation $(\hat{\mathbf{R}}, \hat{\mathbf{t}})$ estimated from the cow42-37 range images leads the SoftICP algorithm to take a long time of 499 seconds to converge to a wrong solution with relative errors of the rotation axis, rotation angle and translation vector as large as 162.02%, −57.24 and 172.59% respectively. In contrast, the relatively accurate transformation $(\hat{\mathbf{R}}, \hat{\mathbf{t}})$ from the bunny120-60 range images produces approximately 10 times smaller error of about 15%. This result demonstrates that it is feasible for FEM methods to align overlapping range images coarsely and that the SHOT algorithm is quite effective for the FEM tasks. The experiments further show that despite the fact that these images were captured under typical imaging conditions and the objects include free form shapes with varied complexities of geometry, varied degrees of overlap, varied magnitudes of transformation, and varied levels of imaging noise, reasonable registration results can still be obtained from these range images.

5.2 Point match evaluation

Despite the fact that FEM methods have been seen capable to provide good alignments between overlapping range images, the estimated underlying transformations are normally needed to be refined using a variant of ICP algorithms [32]. The more accurate the initial transformation $(\hat{\mathbf{R}}, \hat{\mathbf{t}})$ is, the more likely the ICP variants enhance the accuracy of the FEM. In this section, the use of the point matches for better estimation of the underlying transformation is experimentally investigated. This result also emphasizes the significance of the point matches which impose influential effects on the estimated transformation. Experimental results to highlight these points can be seen in Figure 11 and Tables 1 and 2.

It is also noted from Figure 11 that the alignment accuracy of the transformations $(\hat{\mathbf{R}}, \hat{\mathbf{t}})$ has been enhanced by the RANSAC algorithm with a large amount of interpenetration [38] for the cow42-37 images, but has been worsened for the bunny120-60 images, as demonstrated by the fact that the two images have been more significantly displaced in the 3D space. These observations have been verified by Table 2, bringing the relative errors in the rotation axis, rotation angle and translation vector of the underlying transformation from approximately 170% down to as small as 10% over the cow42-37 images, but increasing those from about 16% to about 20%. These results do reveal the difficulty for the RANSAC method to accurately classify the established PPMs into two categories: inliers and outliers and measure the quality of the candidate underlying transformations. In contrast, the proposed NEP method successfully bring all the overlapping range images into better alignment with less displacement and more interpenetration, with errors as small as 5% over the cow42-37 images and about 10% for the bunny120-60 images. These results show that the proposed NEP method is more powerful in evaluating the false positives highly corrupted PPMs and thus producing more accurate underlying transformations for the alignment of the overlapping range images. The excellent performance of the proposed NEP algorithm indicates that the FEM perhaps may not be as challenging an issue as expected in range image research. It may open a novel avenue for the analysis of the range images. This means that the development of either the FEM or PPMs evaluation method may lead to accurate alignment of overlapping range images that may satisfy the requirements of other related tasks such as object modeling, classification, and recognition and simultaneous localization and map building (SLAM).

6 Conclusions

In this article, we have discussed the characteristics of range images generated by several different types of range cameras as an introduction for 3D model imaging. Unlike digital broadband imaging, which only encodes the reflectance property of the scene, range images encode the distance of the scene from a reference point or plane instead, depending on the type of the range camera. Since range images provide geometrical information of the scene, the technology is important to numerous applications in the real world such as digital archiving of historical relics, simultaneous localization and mapping for autonomous navigation and quality assurance for industrial manufacturing as outlined above in Section 1.2.

Due to the narrow field of view in range cameras, a number of range images recorded at different viewing points are needed to be stitched together to cover the scene or the surface of the object of interest. The first step is to align the range images in a common coordinate system. Feature extraction and matching methods are usually needed for this kind of range image registration. To this end, we have outlined one of the most widely exploited methods, the signature of histograms of orientations (SHOT) algorithm [26], for the establishment of point matches between a given pair of overlapping range images. However, range images are often corrupted by imaging noise and cluttered by background, and objects include relatively simple geometry (such as plane, sphere, and cylinder), the matched points normally consist of up to 90% false

positives. Such large number of false matched points make registration of overlapping range images a very challenging task. It is critically important to establish a robust methodology for the identification of false positive matched points.

Inspired by the gentle adaptive boosting method [28], the RANSAC method [27] has been widely adopted for the image marching task [26], in this article, an alternative method based on the normalized error penalization (NEP) approach to complement the RANSAC method with an added advantage of easy implementation has been outlined. The method penalizes all those matched points with errors far away from the majority points. Then the transformation based on these "good" matches is estimated in the weighted least squares sense and the result is compared with that of the ground truth data. It is observed that the errors in the rotation axis, rotation angle and translation vector of the underlying transformation is reduced from approximately 170% down to 10% using the RANSAC algorithm and it is further reduced to 5% when the proposed NEP algorithm is applied. This is an encouraging result which may even be acceptable for some real world applications as outlined above in Section 1.2 without the need to be refined by the traditional ICP algorithm.

The readers are suggested to consult reference [32] for further details of the de facto standard registration technique, iterative closest point (ICP) algorithm. Readers who are interested in the variants of this algorithm may find the following papers useful: [39, 35, 40, 41, 42]. To gain some ideas about seminal work in feature extraction and matching, it is recommended to read references [33, 43]. For advanced readers who are interested in enhancing the FEM algorithm, the random sample consensus (RANSAC) algorithm [27] may be useful as the first step to estimate the point matches and the underlying transformation. To learn the classical techniques for the integration of the registered range images, the readers are suggested to refer to [1, 2, 3].

References

[1] B. Curless, M. Levoy. A volumetric method for building complex models from range images, in *SIGGRAPH*, pp. 303-312, 1996.

[2] G. Turk, M. Levoy. Zippered Polygon Meshes from Range Images, in *SIGGRAPH*, pp. 311-318, 1994.

[3] R.A. Newcombe, S.Izadi, O. Hilliges, D. Molyneaux, D. Kim, A.J. Davison, P. Kohi, J. Shotton, S. Hodges, A. Fitzgibbon. KinectFusion: Real-time dense surface mapping and tracking, in *Proc. 2011 10th IEEE International Symposium on Mixed and Augmented Reality*, 2011.

[4] USF range image database, http://marathon.csee.usf.edu/range/DataBase.html

[5] K. Storjohann. Laser range camera modeling. *Technical Report ORNL/TM-11530*, Oak Ridge National Laboratory, Oak Ridge, Tennessee, 1990.

[6] O.H. Dorum, A. Hover and J.P. Jones. Calibration and control for range imaging in mobile robot navigation, in *Proceedings of International Conference on Vision Interface*, 1994, pp. 25-32.

[7] A. Hoover. Chapter 4: Range Cameras. *The Space Envelope Representation for 3D Scenes*, PhD Dissertation, University of South Florida, 1996.

[8] K. Lai, L. Bo, X. Ren, and D. Fox. A Large-Scale Hierarchical Multi-View RGB-D Object Dataset, in *IEEE International Conference on Robotics and Automation (ICRA)*, 2011, pp. 1817-1824.

[9] I.S. Kweon, R. Hoffman, E. Krotkov. Experimental Characterization of the Perceptron Laser Rangefinder, *Performance organization report number CMU-RI-TR-91-01*, The Robotics Institute, Carnegie Mellon University, 1991.

[10] Kinect for Windows Sensor Components and Specifications. https://msdn.microsoft.com/en-us/library/jj131033.aspx

[11] C.D. Mutto, P. Zanuttigh, G.M. Cortelazzo. Time-of-Flight Cameras and Microsoft Kinect^{TM}: A user perspective on technology and applications, Springer, 2013.

[12] OSU(MSU/WSU) range image database. http://sampl.ece.ohio-state.edu/data/3DDB/RID/index.htm.

[13] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. International Journal of Computer Vision, **2002**, 47(1/2/3), pp. 7-42.

[14] Y. Furukawa, J. Ponce. Accurate, dense and robust multi-view stereopsis. *IEEE Trans. PAMI*, **2010**, 32(8), pp. 1362-1376.

[15] The Stanford 3D Scanning Repository. http://graphics.stanford.edu/data/3Dscanrep/

[16] H. Cantzler. An overview of range images. http://homepages.inf.ed.ac.uk/rbf/CVonline/ LOCAL_COPIES/CANTZLER2/range.html

[17] Range imaging. https://en.wikipedia.org/wiki/Range_imaging

[18] Range images and range grids. https://courses.cs.washington.edu/courses/ cse558/01sp/project2/description/details.html

[19] M. Levoy, K. Pulli, et al. The Digital michelangelo project: 3D scanning of large statues, in *Proceedings of SIGGRAPH*, 2000, pp. 131-144.

[20] A. Banno, T. Masuda, T. Oishi, K. Ikeuchi. Flying laser range sensor for large-scale site-modeling and its application in Bayon digital archival project. *International Journal of Computer Vision*, **2008**, 78, pp. 207-222.

[21] R.B. Rusu, Z.S. Marton, N. Lodow, M. Dolha, M. Beetz. Towards 3D Point cloud based object maps for household environments. *Robotics and Autonomous Systems*, **2008**, 56, pp. 927-941.

[22] J. Liu, A. Jakas, A. Al-Obaidi, Y. Liu. Practical issues and development of underwater 3D laser scanners, in *Proceedings of 15th IEEE International Conference on Emerging Technologies and Factory Automation*, 2010, pp. 1-8.

[23] A. Hilton, A.J. Stoddart, J. Illingworth, T. Windeatt. Automatic inspection of loaded PCBs using 3D range data, in *Proceedings of SPIE 2183, Machine vision applications in industrial inspection*, 1994, pp. 226-237.

[24] The open source system for processing and editing 3D triangular meshes, http://www.meshlab.net/

[25] Java SE Desktop Technologies: Java 3D API. ttp://www.oracle.com/technetwork/articles/javase/index-jsp-138252.html

[26] F. Tombari, S. Salti, L.D. Stefano. Unique signatures of histograms for local surface description, in *Proc. ECCV*, 2010, pp. 347-360.

[27] M.A. Fishcler, R.C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of ACM*, **1981**, 24(6), pp. 381-295.

[28] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting. *The Annals of Statistics*, **28(2)**, pp. 337–374, 2000.

[29] J.L. Bentley (1975). Multidimensional binary search trees used for associative searching, *Communications of the ACM*, **18**(9): 509-517, 1975.

[30] W.E. Lorensen, H.E. Cline. Marching Cubes: A High Resolution 3D Surface Construction Algorithm, *Computer Graphics*, **21**(3): 163-169, 1987.

[31] Y. Guo, M. Bennamoun, F. Sohel, M. Lu, J. Wan, N.M. Kwok. A Comprehensive Performance Evaluation of 3D Local Feature Descriptors. *IJCV*, **116**, pp. 66-89, 2016.

[32] P. J. Besl, N. D. McKay. A method for registration of 3D shapes. *IEEE Trans. PAMI*, **14**: 239-256, 1992.

[33] H. Chen and B. Bhanu. 3D free form object recognition in range images using local surface patches, *Pattern Recognition Letters*, **2007**, 28, pp. 1252-1262

[34] G. Taubin. Estimating the tensor of curvature of a surface from a polyhedral approximation, in *Fifth International Conference on Computer Vision (ICCV'95)*, 1995, pp. 902-907.

[35] Y. Liu. Automatic registration of overlapping 3D point clouds using closest points. *Image and Vision Computing*, **2006**, 24, pp. 762-781.

[36] S. Gold, A. Rangarajan, et al. New algorithms for 2-D and 3-D point matching: pose estimation and correspondence. *Pattern Recognition*, **1998**, 31, pp. 1019-1031.

[37] C. Torre-Ferrero, J.R. Llata, S. Robla, E.G. Sarabia. A similarity measure for 3D rigid registration of point clouds using image-based descriptor with low overlap, in *Proc. IEEE 12th Int. Conf. on Computer Vision Workshops*, 2009, pp. 71-78.

[38] L. Silva, Olga R.P. Bellon, and K.L. Boyer. Precision range image registration using a robust surface interpenetration measure and enhanced genetic algorithms. *IEEE Trans. PAMI*, **2005**, 27, pp. 762-776.

[39] G. Dewaele, F. Devernay, and H. Horaud. Hand motion from 3D point trajectories and a smooth surface model, in *Proc. ECCV*, 2004, pp. 495-507.

[40] J.M. Phillips, R. Liu, C. Tomasi. Outlier robust ICP for minimizing fractional RMSD, in *Proc. 3DIM*, 2007, pp. 427-434.

[41] K. Pulli. Multiview registration for large data sets, in *Proc. 3DIM*, 1999, pp. 160-168.

[42] S. Rusinkiewicz, M. Levoy. Efficient variants of the ICP algorithm, *Proc. Third International Conference on 3-D Digital Imaging and Modeling*, 1999, pp. 145-152.

[43] A. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3D scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, **1999**, 21, pp. 433-449.